caBIG cancer Biomedical Informatics Grid

Challenge of Annotation Exchange:
a developers perspective

TrAPSS

Transcript Annotation Prioritization and Screening System

Center for Bioinformatics and Computational Biology

THE UNIVERSITY OF IOWA

Dr. Terry Braun

NATIONAL CANCER INSTITUTE

Outline

- Problem statement
- Quick overview of standards/protocols
- Our (non-caBIG compatible) solution
- Examples
- Conclusions/Issues

Problem Statement

- Goals of TrAPSS – Accelerate Mutation Identification
- Automate
  - disease gene identification and mutation screening
  - acquisition of gene structure and genomic context
  - identification of domains, secondary structures, SNPs, repeats, cross-species homologies
  - prioritization of genes
  - prioritization of sub-regions of genes
  - selection of assay reagents (ex. primers)
  - data management (ex. track genes, primers, annotation)
- Scientifically: Determine which types of information and heuristics may improve the ability to predict regions of genes that would be more likely to harbor phenotype-altering variations

Needs and Issues

- How do we store/retrieve annotation?
- How do we categorize annotation?
- What annotations?
- From where?
- What if it changes?
- Do we mirror or cache?

- Examples: intervals, gene lists, expression (hybridizations, tissues, ESTs, SAGE), pathways, regulatory elements, literature…

Existing Standards / Protocols

- http://obo.sourceforge.net/
  - Open Biological Ontologies is an umbrella web address for well-structured controlled vocabularies for shared use across different biological domains.
  - 43

- eVOC

- http://www.sanbi.ac.za/evoc/
  - set of orthogonal controlled vocabularies that unifies gene expression data by facilitating a link between the genome sequence and expression phenotype information

caBIO

http://ncicb.nci.nih.gov/core/caBIO

The cancer Bioinformatics Infrastructure Objects (caBIO) model and architecture is the primary programmatic interface to caCORE. The heart of caBIO is its domain objects, each of which represents an entity found in biomedical research. These domain objects are related to each other, and examining these relationships can bring to the surface biomedical knowledge that was previously buried in the various primary data sources.

Java API, SOAP, HTTP-XML, Perl API

The caBIO software development process is an iterative software development approach that leverages a combination of elements from the Rational Unified Process (RUP) and eXtreme Programming (XP). Use case models are created by utilizing the domain expertise available at the Center for Bioinformatics to evaluate existing projects and investigate industry standards.

Once the use case analysis is completed, an iterative functional design and development process is applied, which allows for rapid and segmented development of the application. During an iteration, all of the software development activities are executed. The artifacts associated with each functional iteration include: detailed use cases describing the function; class and sequence diagrams; a system architecture diagram; the actual software code; a published API; a project plan describing subsequent iterations; and a test plan for software validation.

Issues for Developers: familiarity with this process, overhead, training

cancer Data Standards Repository (caDSR)
http://ncicb.nci.nih.gov/core/caDSR

- CDE (common data elements)
  - Object, class, data structure, …
  - promote the efficient sharing and interpretation of information
- SNP, mutation, domain, gene, transcript, exon
- No data elements matching the search criteria found.
- Developer issues: my “data element” does not exist

Existing Standards/Protocols

- DAS
- http://biodas.org/
- client-server system in which a single client integrates information from multiple servers. It allows a single machine to gather up genome annotation information from multiple distant web sites, collate the information, and display it to the user in a single view. Little coordination is needed among the various information providers.

DAS view of annotation

Annotation - An entity which:

- 1. Is anchored to the genome map via a stop and start value relative to the reference subsequence;
- 2. Possesses an ID unique to the server and a structured description of its nature and attributes;
- 3. Optionally associated with Web URLs providing human-readable information about the annotation (via link);
- 4. Possesses types, methods, and categories.

Annotation Ontology

- The more we examined existing ontologies, the more we decided we needed a custom solution
- Developed our own “annotation ontology” -- albeit very simple
- Type (7 classes)
  - literature, x_ref, seq_feature, expression, pathway, mapping, function
- sec_typ
- value

Annotation Ontology (Use Cases)

Type	sec_type	value
Literature	-> ref	-> pubmed/omim
Xref	-> unigene	
	-> geneCard	
Seq_feat	-> sequence	
	-> cds_start	-> Ensembl
	-> polyAsite	
	-> SNP	-> dbSNP
	-> exon_start	
	-> exon_stop	
	-> structure	-> NNpredict

Annotation Ontology

Mapping	-> BAC	
	-> RH	-> GeneMap
	-> cytogenetic	-> UniGene
	-> FISH	
	-> Genetic	
	-> genomic	-> UCSC
Function	-> GO	
Pathway	-> WNT signaling	-> KEGG
	-> prot-inter	-> BIND

Annotation Ontology

Expression

-> signal

-> change p-value

-> change expression

-> position on slide

-> SAGE

-> cDNA

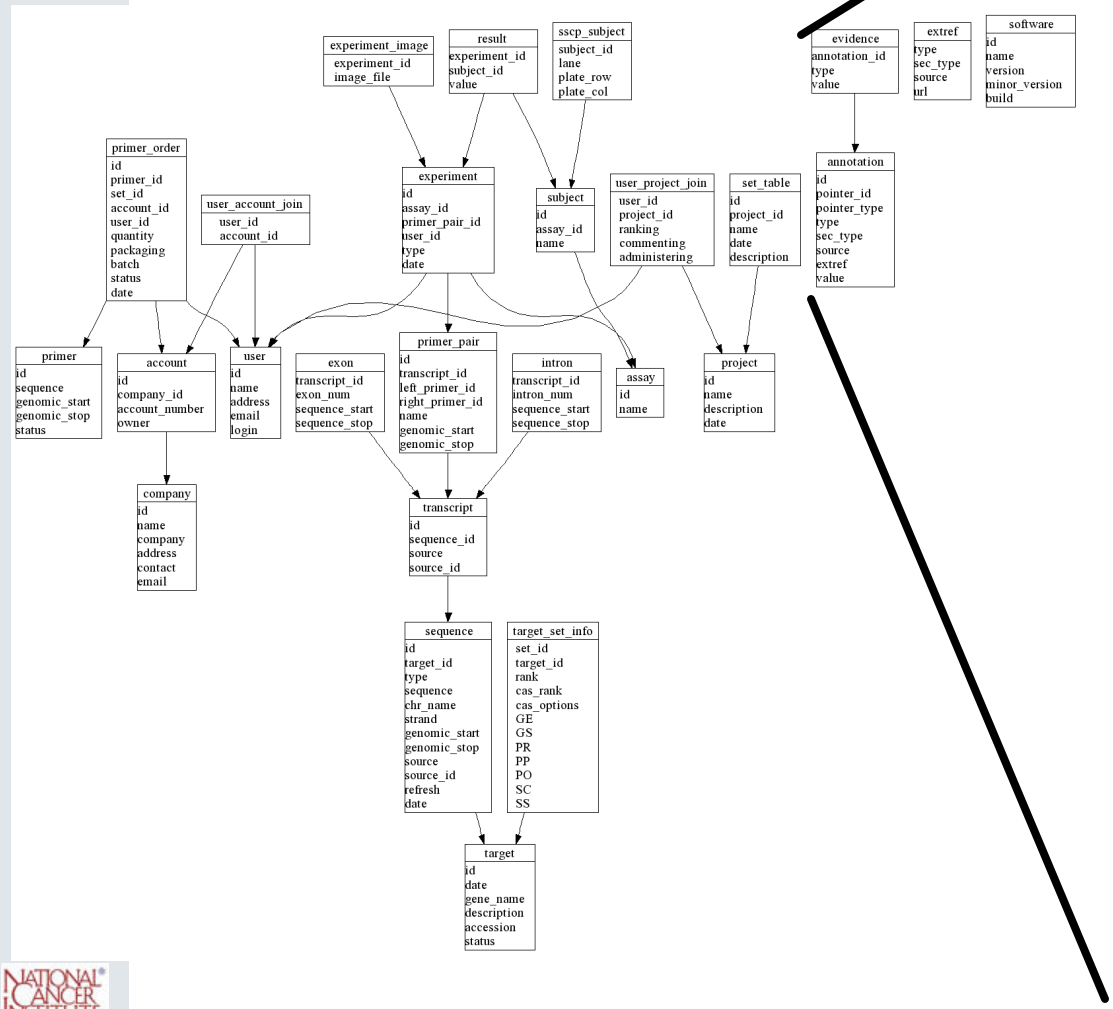
-> Northern

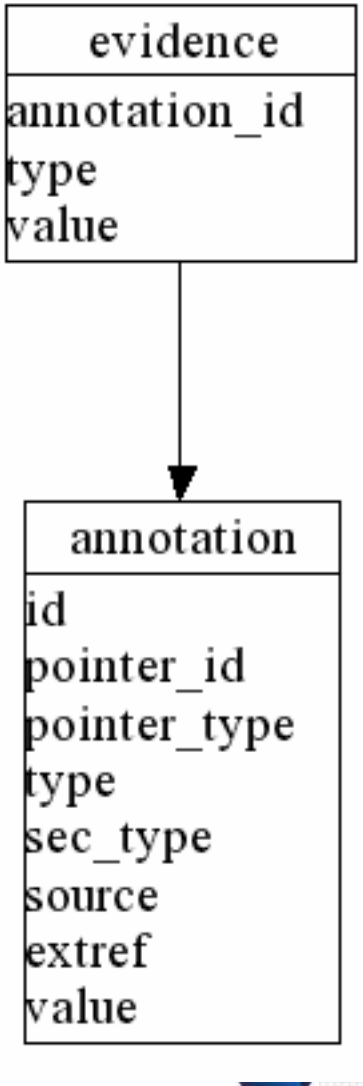
-> cluster size

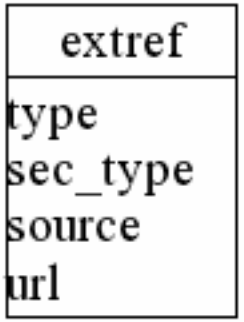
SQL


```
create table extref    (type varchar(32) not null,
                        sec_type varchar(32),
                        source varchar(32),
                        url varchar(255)) type=innodb;
create table annotation
(id bigint unsigned not null auto_increment primary key,
                        pointer_id bigint unsigned not null,
                        pointer_type enum('transcript', 'target', 'sequence') not null
                       default 'transcript',
                        type varchar(32) not null,
                        sec_type varchar(32),
                        source varchar(32),
                        extref varchar(32),
                        value varchar(255),
                        index (id),
                        index (pointer_id)) type=innodb;
create table evidence  (annotation_id bigint unsigned not null,
                        type varchar(32) not null,
                        value text not null,
                        index (annotation_id),
                        foreign key (annotation_id) references annotation(id)
                        on delete restrict on update restrict) type=innodb;
```

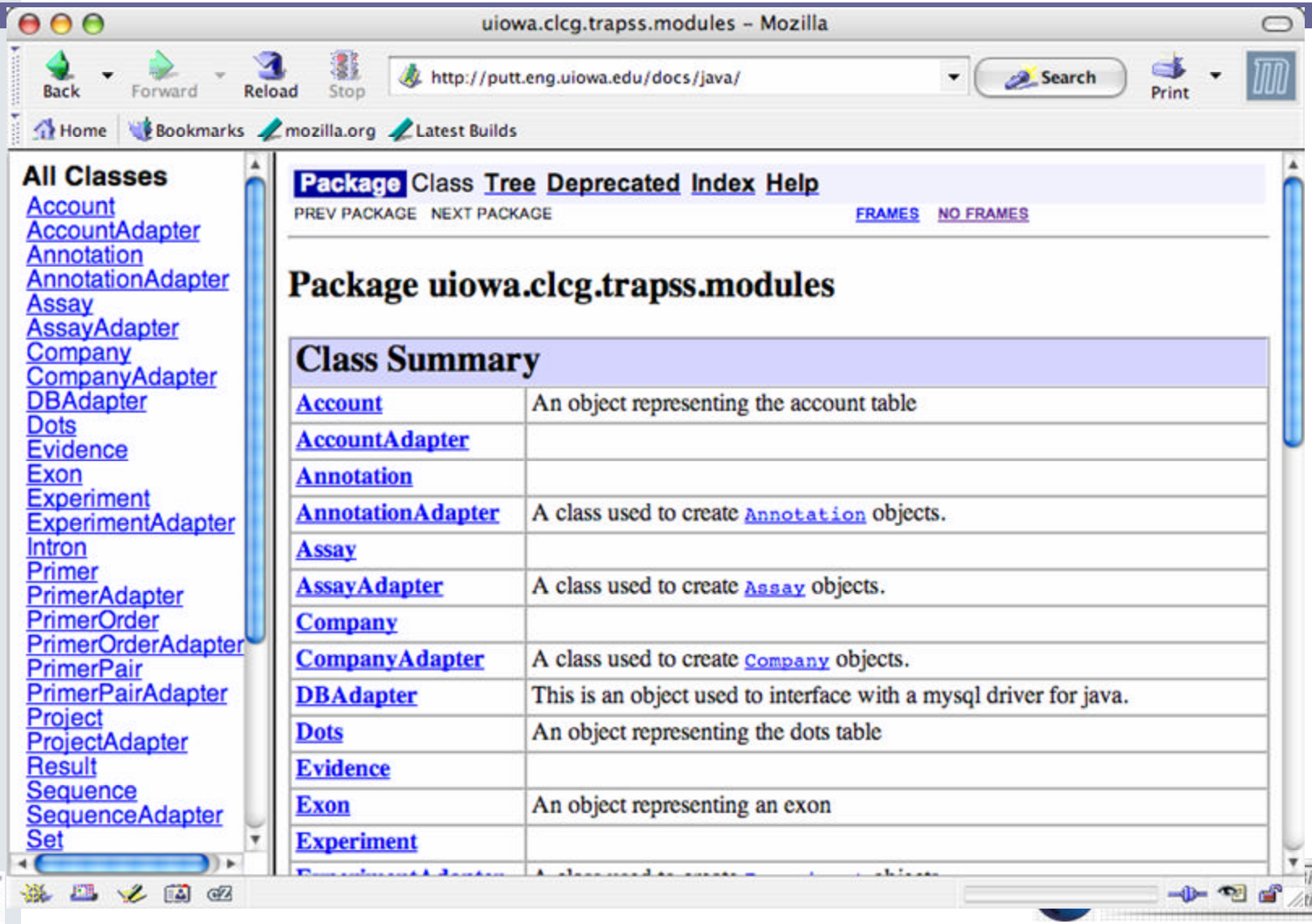
Subset of Database

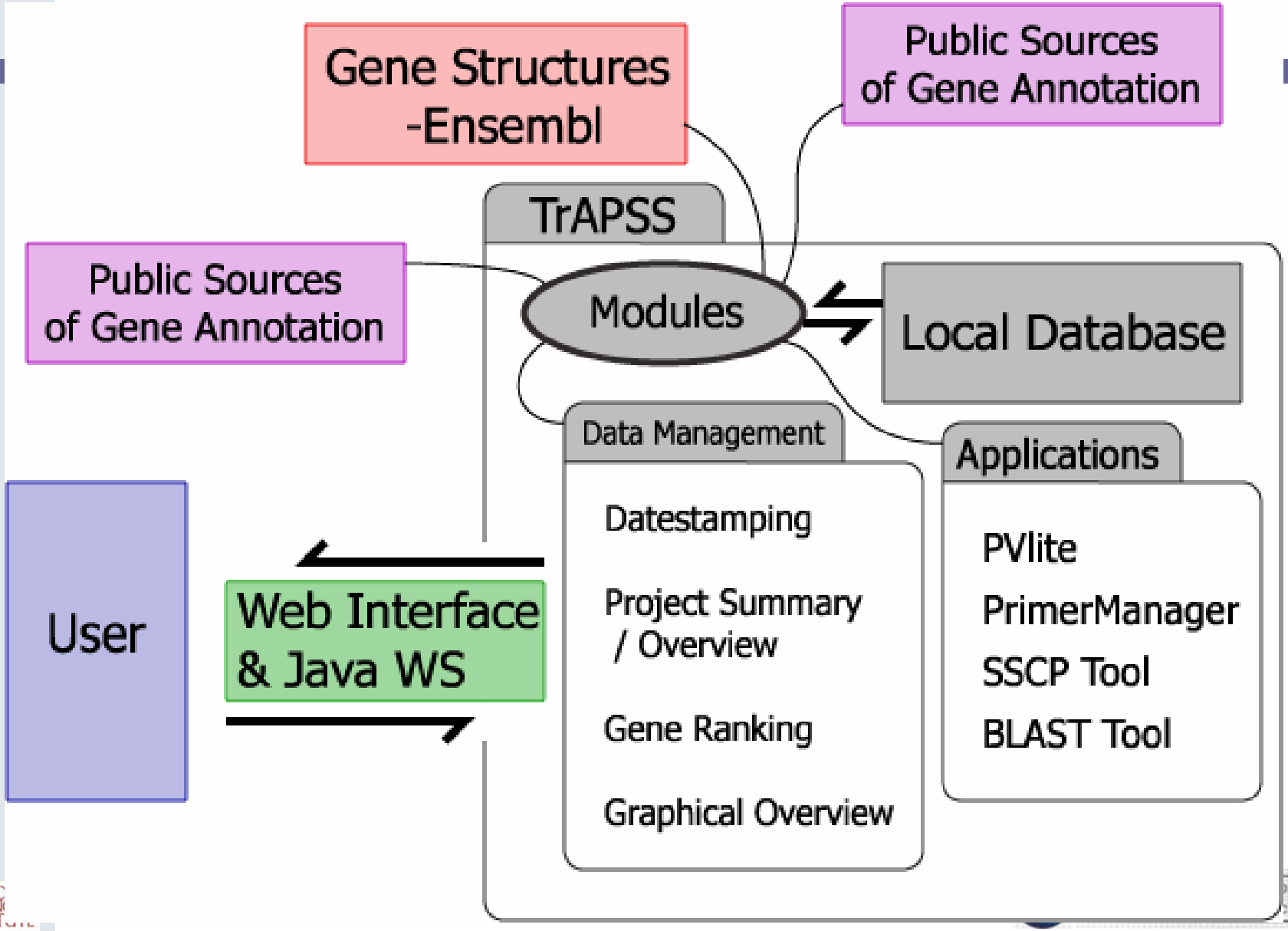






Java API


uiowa.clcg.trapss.modules - Mozilla
Back
Forward
Reload
Stop
http://putt.eng.uiowa.edu/docs/java/
Search
Print
Home
Bookmarks
mozilla.org
Latest Builds
All Classes
Account
AccountAdapter
Annotation
AnnotationAdapter
Assay
AssayAdapter
Company
CompanyAdapter
DBAdapter
Dots
Evidence
Exon
Experiment
ExperimentAdapter
Intron
Primer
PrimerAdapter
PrimerOrder
PrimerOrderAdapter
PrimerPair
PrimerPairAdapter
Project
ProjectAdapter
Result
Sequence
SequenceAdapter
Set
Package Class Tree Deprecated Index Help
PREV PACKAGE NEXT PACKAGE
FRAMES NO FRAMES
Package uiowa.clcg.trapss.modules
Class Summary
Account
An object representing the account table
AccountAdapter
Annotation
AnnotationAdapter
A class used to create Annotation objects.
Assay
AssayAdapter
A class used to create Assay objects.
Company
CompanyAdapter
A class used to create Company objects.
DBAdapter
This is an object used to interface with a mysql driver for java.
Dots
An object representing the dots table
Evidence
Exon
An object representing an exon
Experiment


Gene Structures
-Ensembl
Public Sources
of Gene Annotation
TrAPSS
Public Sources
of Gene Annotation
Modules
Local Database
Data Management
Applications
User
Web Interface
& Java WS
Datestamping
Project Summary
/ Overview
Gene Ranking
Graphical Overview
PVlite
PrimerManager
SSCP Tool
BLAST Tool

Conclusions

- Observations
  - Were not concerned about data exchange (out)
  - Faster to develop our own vocabulary/objects than to modify any existing standards and protocols
  - Decision came down to the availability of Ensembl (bio-perl-like) modules that greatly streamlined our ability to extract genomic data and annotation
  - Modeled our Java, PHP, and perl API's after Ensembl's
- Developers perspective (TrAPSS Project -- cost today)
  - Considerable effort to adopt standards and protocols within scope of caBIG
  - Overhead of caCORE, CDEs (assuming these are the frameworks)
  - Most significant perceived hurdles
    - Learning the architecture - caCORE, CDEs, vocabulary
    - Contributing to the architecture
    - Re-implementing the code
  - Do the standards and protocols currently exist for "annotation exchange" ???